



Source Camera Device Identification from Videos

Guru Swaroop Bennabhaktula^{1,2,3} · Derrick Timmerman¹ · Enrique Alegre^{2,3} · George Azzopardi¹

Received: 14 June 2021 / Accepted: 12 May 2022 / Published online: 4 June 2022
© The Author(s) 2022

Abstract

Source camera identification is an important and challenging problem in digital image forensics. The clues of the device used to capture the digital media are very useful for Law Enforcement Agencies (LEAs), especially to help them collect more intelligence in digital forensics. In our work, we focus on identifying the source camera device based on digital videos using deep learning methods. In particular, we evaluate deep learning models with increasing levels of complexity for source camera identification and show that with such sophistication the scene-suppression techniques do not aid in model performance. In addition, we mention several common machine learning strategies that are counter-productive in achieving a high accuracy for camera identification. We conduct systematic experiments using 28 devices from the VISION data set and evaluate the model performance on various video scenarios—flat (i.e., homogeneous), indoor, and outdoor and evaluate the impact on classification accuracy when the videos are shared via social media platforms such as YouTube and WhatsApp. Unlike traditional PRNU-noise (Photo Response Non-Uniform)-based methods which require flat frames to estimate camera reference pattern noise, the proposed method has no such constraint and we achieve an accuracy of $72.75 \pm 1.1\%$ on the benchmark VISION data set. Furthermore, we also achieve state-of-the-art accuracy of 71.75% on the QUFVD data set in identifying 20 camera devices. These two results are the best ever reported on the VISION and QUFVD data sets. Finally, we demonstrate the runtime efficiency of the proposed approach and its advantages to LEAs.

Keywords Source camera identification · Sensor pattern noise · Digital video forensics · Constrained ConvNet

This article is part of the topical collection “Pattern Recognition Applications and Methods” guest edited by Ana Fred, Maria De Marsico and Gabriella Sanniti di Baja.

✉ Guru Swaroop Bennabhaktula
g.s.bennabhaktula@rug.nl

Derrick Timmerman
d.k.timmerman@rug.nl

Enrique Alegre
enrique.alegre@unileon.es

George Azzopardi
g.azzopardi@rug.nl

¹ Information Systems Group at the Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence, University of Groningen, Groningen, The Netherlands

² Group for Vision and Intelligent Systems at the Department of Electrical, Systems, and Automation, University of León, León, Spain

³ Researcher at INCIBE (Spanish National Cybersecurity Institute), León, Spain

Introduction

With the widespread increase in the consumption of digital content, camera device identification has gained a lot of importance in the digital forensics community. Law Enforcement Agencies (LEAs) have a special interest in the developments in this field as the knowledge of the source camera, extracted from the digital media, can provide additional intelligence in the fight against child sexual abuse content. Our work is part of the EU-funded 4NSEEK project¹ which is aimed at the development of cyber-tools to assist LEAs in identifying the source of illicit content involving minors.

Most of the research in camera identification has been limited to the investigation of digital images generated by a camera device. In contrast, Source Camera Identification (SCI) based on videos has not seen much progress. An application of video forensics can be seen by considering the following real-world scenario. When LEAs have a warrant to conduct a search in the properties of alleged offenders, for any device that they find with an in-built video camera, they

¹ <https://www.incibe.es/en/european-projects/4nseek>

can take multiple random videos, which can then be used as reference/training samples to learn the proposed approach. Therefore, the proposed approach can be reconfigured with every new device that LEAs find. Subsequently, any suspicious video files can be processed by the proposed method to determine if they were captured by one of the known devices or not. An additional follow-up study would be to determine whether two or more videos were originated from the same device, without having to know the specific device that was used. This would help LEAs to link multiple cases to the same offender, for instance.

Another interesting application of video forensics is to identify copyright infringements of digital videos. This is relevant in the current scenario where a lot of copyright content is freely available on online platforms like YouTube.

A major challenge in SCI is to mitigate the impact of the presence of scene content while extracting camera traces from images or videos. The presence of scene content makes the extraction of camera noise quite difficult as state-of-the-art methods, such as convolutional neural networks (ConvNets), tend to learn the details from the scene rather than the camera noise. To address this problem, Bayer et al. [6] proposed to use a constrained convolutional layer as the first layer of a ConvNet. The constraints imposed on the convolutional filters are aimed to increase robustness at extracting the camera noise by suppressing the scene-level details. A limitation of their approach is based on the requirement of the input image to be monochrome or a single-channel image. As most of the digital media is generated as color or multi-channel, by converting them to monochrome, the native information that is present in each individual color channel is lost. To overcome this problem, an extended version of the constrained convolutional layer that can handle multi-channel inputs was proposed in [52]. These methods demonstrate the usefulness of constrained convolutional layer for shallow ConvNets. In our work, we demonstrate that such constraints are not needed when using deep ConvNets such as MobileNet, and ResNet.

The key contributions of this paper are threefold. First, we evaluate ConvNets with increasing levels of sophistication to understand the relation between network complexity and its corresponding performance to source camera identification. Second, we show that neither the constrained convolutional layer nor the residual noise images are necessary when sophisticated ConvNets are employed. Additionally, we mention a few common algorithm choices that could be detrimental to camera identification from videos. Third, we set a benchmark on the VISION data set using 28 camera devices (see Table 1) that can be used to evaluate new methods for SCI on the VISION data set. The selection of these devices ensures that multiple instances of the same camera model are always included in the data set, which allows us to test the performance at the device level. Moreover, to the

best of our knowledge, this is the largest subset of devices from the VISION data set used to conduct experiments for SCI using only videos and trained using a single model. Furthermore, we also set a new benchmark result on the QUFVD data set [1] using all the 20 camera devices. Finally, we demonstrate the runtime efficiency of our algorithm during deployment which becomes necessary to conduct forensic investigations in time critical situations.

We also share the source code² for further dissemination of our approach and experiments.

The rest of the paper is organized as follows. The next section gives an account of the state-of-the-art on SCI using videos. In the subsequent section, we describe the proposed technique, followed by the experimental details. A brief discussion of the results is elucidated next, which is finally followed by our conclusions.

Related Works

Source camera identification has become a topic of interest after the widespread popularity of digital cameras [32, 33]. Most of the literature in this area is concerned with the investigation of digital images and not involving videos. With digital videos becoming increasingly accessible and popular due to the advances in camera technology and the internet, it has now become crucial to investigate SCI using videos. We begin by reviewing the relevant literature from the investigation of digital images as these methods are closely related to video-based SCI.

When the same scene is captured by two different digital cameras at the same time and under the same conditions, the final images that are generated are never exactly the same. There are always a few visually noticeable differences, such as in color tones, radial distortions, and image noise, among others. Such variations are perceptible in images generated from different camera models. This is due to the fact that every camera model has a different recipe for image generation, which is commonly referred to as the camera pipeline. A typical camera pipeline consists of optical lenses, anti-aliasing filter, color filter arrays, imaging sensor, demosaicing, and post-processing operations as depicted in Fig. 1. When the light from the scene enters the lenses, it gets processed by a sequence of hardware and software processing steps before the generation of the final digital image. As the implementation of the camera pipeline is distinct for each camera model the final image generated by them is also unique. Even though digital images captured by the camera devices of the same model type, when examined closely they

² <https://github.com/bgswaroop/scd-videos>.

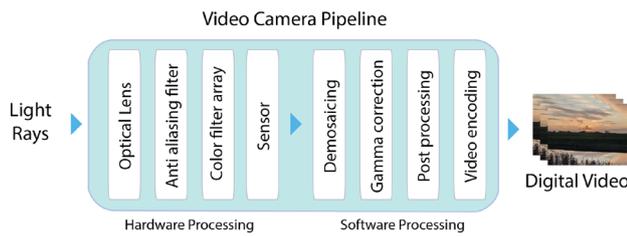


Fig. 1 Video generation pipeline inside digital cameras. The light from the scene is continuously sampled by the digital camera at a pre-determined frequency to generate digital video frames. The camera pipeline typically involves a series of lenses, an anti-aliasing filter, color filter arrays, followed by the imaging sensor. These hardware processing steps are succeeded by a set of software processing steps consisting of demosaicing, video encoding, and other post-processing operations to generate the output digital video

are not exactly the same. This subtle variation is due to the unique sensor pattern noise generated by every camera sensor, which makes it possible to identify individual camera devices of the same camera model. In this work, we consider the problem of camera device identification from videos.

Kurosawa et al. [32, 33] in their initial study on sensor noise, have observed the noise pattern generated by the dark currents and reported that such noise is unique to every camera sensor. Their experiments were conducted on nine camera devices using videos generated by the involved cameras on a flat scene. Hundred video frames were extracted from each video to determine the noise pattern of dark currents. Their study laid the foundation for further experiments concerning camera identification. That approach [33], however, is limited as it relies on flat video frames and requires access to physical camera device. To overcome the restriction using only flat frames, Kharrazi et al. [30] proposed to extract 34 hand-crafted features from images and showed that those features enabled them to identify the source camera device from natural images.

As shown in Fig. 2, the sensor noise can be categorized into shot noise and pattern noise. Shot noise is a stochastic random component that is present in every image and can be suppressed by frame averaging. The resulting noise which survives frame averaging is defined as the pattern noise [23], which can be further classified into Fixed Pattern Noise (FPN) and Photo-response non-uniformity Noise (PRNU). FPN is generated by the dark currents when the sensor is not exposed to any light. PRNU is generated when the sensor is exposed to light and is caused due to different sensitivities of pixels to incoming light intensity. Unlike earlier methods [32, 33] that rely on FPN, Lukas et al. [38] in their seminal work showed that it is possible to identify source camera devices by extracting PRNU noise from images. They determined the sensor pattern noise by averaging the noise obtained from multiple images using a denoising filter. In particular, they used a wavelet-based denoising filter to

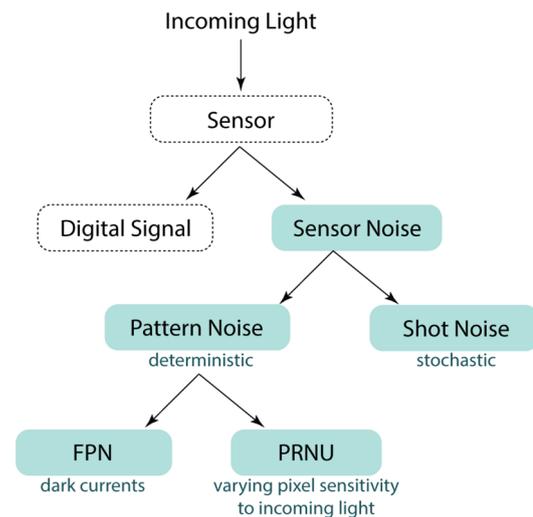


Fig. 2 Noise classification in imaging sensors. The noise generated by an imaging sensor (sensor noise) can be classified into a random component, namely shot noise, and into a deterministic component commonly referred to as (sensor) pattern noise. The pattern noise is further classified into FPN and PRNU based on the incoming light intensity. FPN is generated when the scene is dark and PRNU otherwise

compute the noise residuals. The authors show that using such a filter before frame averaging helps in suppressing scene content. This idea is similar to more recent works [6, 52] that allow the neural networks to learn constrained filters for scene suppression.

Other approaches have been presented that target to enhance the sensor pattern noise [34, 36, 41]. Such methods are based on the idea proposed in [38], where a handcrafted denoising filter was used to extract camera noise. All the methods mentioned thus far target the sensor noise or the noise generated by the imaging sensor. As shown in Fig. 1, a typical camera pipeline also consists of other processing steps. Therefore, methods to identify the source camera based on the artifacts resulting due to the CFA, demosaicing operations [8, 13, 14, 49], and image compression [2, 15] were proposed. Those techniques, which target to extract the noise from a single processing step, undesirably miss out on the noise patterns generated by a combination of such steps.

In recent years, deep learning-based approaches were proposed for SCI that target a specific step in the camera pipeline. Examples include detection of forgeries involving image in-painting [54, 55], image resizing [12], median filtering [29, 51], and identifying JPEG compression artifacts [3, 4], and so on. Such approaches were shown to be more robust than methods which rely on computing handcrafted features. Though the above methods are based on deep learning, they may not cater to the camera noise generated from the remaining processing steps. Furthermore, deep learning based methods were also proposed [5, 9, 10, 42] to extract



Fig. 3 An overview of the proposed pipeline for source camera identification from videos. It consists of three major steps, namely frame extractor, ConvNet-based classifier, and finally an aggregation step to determine video-level predictions

camera features accounting for noise generated from each of the processing steps in a camera pipeline. This unified approach to SCI accounts for all sources of camera noise and is also used in our work.

Although it is not within the scope of this work, it is worth mentioning that the techniques used for SCI can also be used to address related forensic tasks such as image forgery detection [11, 16, 35], for identifying image manipulation to help fighting against fabricated evidence and fake news. Image forgery detection can be performed as a first step before performing camera identification. Such a combination can make SCI more robust in practice.

Convolutional neural networks (ConvNets) have the ability to learn high-level scene details [31], however, in SCI we need the ConvNets to ignore the high-level scene features and learn to extract features from the camera noise. Chen et al. [14] noticed this behavior when they trained ConvNets for detecting median filtering forgeries. To suppress scene content, they used residual images of median filtering to train their ConvNets. This new approach of suppressing scene content resulted in improved accuracy, which led to the development of two related methods for scene suppression. Firstly, methods that use predefined high-pass filtering to suppress high-level scene content [45]. Secondly, methods that use constrained convolutions [5, 52], which consist of trainable filter parameters.

Most of the methods for SCI address the problem using images. Very few methods have been proposed that use videos for identification and one such method that is closely related to our work is that of Holster et al. [24]. In their work they trained a ConvNet by discarding the constrained convolutional layer proposed by [6]. That layer was removed as it was not compatible to handle color images or multi-channel inputs. Derrick et al. [52] proposed an extended version of the constrained convolution layer and depict the scenarios in which inclusion of multi-channel constrained layer could be beneficial. All these works show the effectiveness of the constrained convolutional layer when used with shallow ConvNets. In our work, we show empirical evidence that such layers are unnecessary with deep ConvNets, such as MobileNet and ResNet.

Dal et al. [17] proposed a multi-modal ConvNet based approach for camera model identification from video sequences. They combine the visual and the audio signals from a video and show that such an approach would result

in a more reliable identification. In our work, we focus on SCI based only on visual content, also because in practice audio content can easily be replaced or manipulated. In the visual content based ConvNet, Dal et al. [17] pick 50 frames equally spaced in time, and extract 10 patches of size 256×256 pixels followed by patch standardization as part of their pre-processing. Furthermore, a pre-trained EfficientNet [50] was employed for classification.

Methodology

In Fig. 3, we illustrate a high-level overview of the proposed methodology. The input videos are processed in three stages. First, the frames are extracted from a video and are then pre-processed. Second, a frame-level classifier is used to predict the class for each frame. Finally, the frame predictions are aggregated to determine the video-level prediction for the given video. We describe each of these stages in detail in the following sections.

Frame Extraction

As the duration of each video is not fixed, we attempt to extract a fixed number of I-frames from each video that are equally spaced in time. This ensures that every video is equally represented in the frame-level data set irrespective of its duration. Our approach of frame selection is different from that in [48], where the first N frames were used to represent a video. As the consecutive frames in a video share temporal content, the scene content and the camera noise will be highly correlated. Choosing the N consecutive frames strategy is favorable when the scene content is relatively homogeneous and is disadvantageous otherwise. Holster et al. [24] used a frame selection strategy based on the frame types, and extracted an equal number of frames from both the categories. In our experiments, we applied and evaluated two different strategies for frame selection. With the first one, we select up to $N = 50$ I-frames equally spaced in time. I-frames are intra-coded frames and have better forensic traces as they do not have any temporal dependency with adjacent frames. It is, however, not always possible to extract 50 I-frames, therefore in our second strategy we extract $N = 50$ frames equally spaced in time. Furthermore,

we investigate the impact of using a different number of frames per video on camera identification during test time.

We take a center-crop of all the extracted frames and normalize the resulting images to the range [0, 1] by dividing by 255. The dimensions of the center-crop are set to 480×800 and were determined based on the dimensions of the video with the smallest resolution in the VISION data set.

Convolutional Neural Networks

We investigate the performance of the proposed approach by considering two different ConvNet architectures, namely MobileNet-v3-small [25] and ResNet50 [21]. Both architectures are deep and are sophisticated in terms of network design when compared to MISLNet [6]. We use the pre-trained versions of these ConvNets in our training where the pre-training was done on ImageNet. In Sect. 5.3 we show empirical evidence of the benefit in pre-training.

MobileNet

MobileNets [25, 26, 47] are a family of ConvNet architectures that were designed to be deployed to mobile platforms and embedded systems. Though those networks have a low memory footprint and high latency, they are sophisticated and achieve comparable results to the state-of-the-art on the benchmark ImageNet data set [31].

In MobileNet-v1 [26], depth-wise separable convolutions were used instead of conventional convolutional filters. Such a combination of depth-wise and point-wise convolutions reduces the number of parameters while retaining the representative power of ConvNets. Reduction in parameters, in general, allows the network to generalize better to unseen examples, as it becomes less specific.

In MobileNet-v2 [47], the depth-wise separable convolutions were used in conjunction with skip connections [21] between the bottleneck layers along with linear expansion layers. The architecture design was further enhanced to reduce latency and improve accuracy in MobileNet-v3 [25]. The performance gain was achieved using squeeze and excitation [27] blocks along with modified swish non-linearities [46]. In our work, we use the MobileNet-v3-small, as it can be easily deployed in systems with limited resources and offers a high runtime efficiency. Furthermore, we change the input dimensions to $480 \times 800 \times 3$ pixels and the output layer to 28 units, which represent the total number of classes. For further architectural details, we refer the reader to [25].

ResNet

ResNets [21] are another popular deep learning architectures that incorporates skip-connections between convolutional layers. Such connections enable identity mapping and allow

the gradients to freely propagate backwards thereby making the network less prone to the problem of vanishing gradients in a traditional deep ConvNet. These networks were further enhanced with a new residual unit in [22]. In our experiments, we use ResNet50 based on the architecture proposed in [22]. We modify the dimensions of the input layer to be $480 \times 800 \times 3$ pixels and the output layer to 28 units representing the total number of devices in VISION data set.

ConstrainedNet

In contrast to the sophisticated ConvNets described earlier, simple ConvNets with few layers are not robust enough to extract the camera features from natural images. The primary reason for such a lack of robustness is the presence of scene content, which obstructs the extraction of noise from images. To overcome this issue, Bayer et al. [6] proposed a constrained convolutional layer, which aims to suppress the scene content. Unlike traditional approaches [36, 38] that use a pre-determined denoising filter to suppress scene content and extract camera noise, a constrained convolutional layer can be trained to suppress scene details. This layer was originally proposed for monochrome images and later extended to process color images [6]. In this work, we explore if an augmentation with constrained convolutional layer is beneficial to sophisticated networks such as MobileNet and ResNet. The details of these filters are briefly described below.

Relationships exist between neighboring pixels which are independent of the scene content. Such an affinity is caused due to the camera noise and can be learned by jointly suppressing the scene details and learning the relationship between each pixel and its neighbors [6]. Thus, the constrained convolutional filters are restricted to learn the extraction of image noise and are not allowed to evolve freely. Essentially, these convolutional filters act as denoising filters, where for each pixel the corresponding output is obtained by subtracting the weighted sum of its neighboring pixels from itself.

Formally, such errors can be determined by placing constraints on each of the K convolutional filters with weights $\mathbf{w}^{(k)}$, as follows:

$$\mathbf{w}^{(k)}(0, 0) = -1 \quad (1)$$

$$\sum_{m,n \neq 0} \mathbf{w}^{(k)}(m, n) = 1, \quad (2)$$

where $\mathbf{w}^{(k)}(0, 0)$ corresponds to the center value of the filter. These constraints are enforced manually after each weight update step during the backpropagation.

The above formulation of constrained convolutional layer was proposed by Bayer et al. [6] and was designed to process

only grayscale images. This was extended by Derrick et al. [52] to process color inputs by imposing filter constraints on all the three color channels j , as shown below:

$$\mathbf{w}_j^{(k)}(0, 0) = -1 \quad (3)$$

$$\sum_{m,n \neq 0} \mathbf{w}_j^{(k)}(m, n) = 1, \quad (4)$$

where $j \in \{1, 2, 3\}$.

Video-Level Predictions

The source camera device of a video is predicted as follows. First, a set of N frames are extracted from a given video as elucidated in Sect. 3.1. Second, the trained ConvNet is used to classify the frames which results in the source video device predictions for each frame. Finally, all the predictions belonging to a single video are compiled together by means of a majority vote, to determine the predicted source camera device. In Sect. 4, we show the efficacy of this step which significantly improves our results.

Experiments

Data Set—VISION

We use the publicly available VISION data set [48], which consists of images and videos captured from a diverse set of scenes and imaging conditions. The data set comprises a total of 35 camera devices representing 29 camera models and 11 camera brands. Specifically, the data set consists of 6 camera models with multiple instances per model, which enables us to investigate the performance of the proposed approach at the device level.

The data set consists of 648 native videos, in that they have not been modified post their generation by the camera. The native videos were shared via social media platforms including YouTube and WhatsApp and the corresponding social media version of the native videos are also available in the data set. Of the 684 native videos, 644 videos were shared via YouTube and 622 in WhatsApp. While both social media platforms compress the native videos, videos shared via YouTube maintain their original resolutions whereas WhatsApp re-scales the video to 480×848 pixels.

Furthermore, the videos captured from each camera are categorized into three different scenarios—flat, indoor, and outdoor. The flat videos have their scene content relatively homogeneous, such as skies and walls. The indoor scenario refers to videos captured inside indoor locations, such as

office and home. Finally, the outdoor scenario contains videos of gardens and streets. With such diversity in the scene content, the VISION data set acts as a suitable benchmark to evaluate source camera identification.

Camera Device Selection Procedure

Among the 35 camera devices of the VISION data set, 28 devices were selected for our experiments. This selection was based on the following criteria:

- The camera devices must contain at least 18 videos in their native resolution encompassing the three scenarios, namely flat, indoor, and outdoor.
- Furthermore, all the native videos should have been shared via both YouTube and WhatsApp.
- Finally, all devices that belong to a camera model with multiple devices are included too. If this criterium is satisfied then the previous two criteria do not need to be satisfied.

The first two criteria ensure that devices with few videos are excluded. Additionally, this allows us to test the same performance of the device identification when the videos are subjected to compression. An exception is made in the final criterium, where multiple devices from the same make and model are always included. This enables us to test camera identification at the device level. By following these criteria, 29 devices were shortlisted. Furthermore, as suggested [48], we exclude the Asus Zenphone 2 Laser camera resulting in 28 camera devices (shown in Table 1). Having selected the camera devices, in the following section we describe the process of creating a balanced training-test set.

Data Set Balancing

The constraint on the number of videos per device for its selection is motivated from the view of creating balanced training and test sets. This is important to keep the data distribution similar for both training and test to avoid any bias towards majorly represented classes. First, we determined the lowest number of native videos present per camera device, which turned out to be 13. These videos were split between a training and a test set such that 7 videos were present in the training and the remaining 6 in the test. This split further ensures that at least 2 videos from each of the three scenarios (flat, indoor, and outdoor) are present in both train and test (with the exception of D02, where only 1 native-indoor video was available to be included in the test set). Thereby, ensuring that all scenarios are equally represented in the splits. Subsequently, the training and the test splits were augmented with the social media versions (WhatsApp and YouTube) of the corresponding native videos. Thus, the

Table 1 List of 28 camera devices considered for our experiments from a total of 35 devices from the VISION data set

| Sr.no. | Device name | Sr.no. | Device name | Sr.no. | Device name | Sr.no. | Device name |
|--------|---------------------------|--------|---------------------------|--------|---------------------------|--------|--------------------------|
| 1 | D01_Samsung_GalaxyS-3Mini | 8 | D08_Samsung_Galaxy-Tab3 | 15 | D16_Huawei_P9Lite | 22 | D28_Huawei_P8 |
| 2 | D02_Apple_iPhone4s | 9 | D09_Apple_iPhone4 | 16 | D18_Apple_iPhone5c | 23 | D29_Apple_iPhone5 |
| 3 | D03_Huawei_P9 | 10 | D10_Apple_iPhone4s | 17 | D19_Apple_iPhone6Plus | 24 | D30_Huawei_Honor5c |
| 4 | D04_LG_D290 | 11 | D11_Samsung_GalaxyS3 | 18 | D24_Xiaomi_RedmiNote3 | 25 | D31_Samsung_GalaxyS4Mini |
| 5 | D05_Apple_iPhone5c | 12 | D12_Sony_XperiaZ-1Compact | 19 | D25_OnePlus_A3000 | 26 | D32_OnePlus_A3003 |
| 6 | D06_Apple_iPhone6 | 13 | D14_Apple_iPhone5c | 20 | D26_Samsung_GalaxyS-3Mini | 27 | D33_Huawei_Ascend |
| 7 | D07_Lenovo_P70A | 14 | D15_Apple_iPhone6 | 21 | D27_Samsung_GalaxyS5 | 28 | D34_Apple_iPhone5 |

three versions of the same video content occur in either of the sets but not in both. This ensures that the evaluation is not influenced by the in-advert classification of the scene content. This scheme resulted in a total of 588 videos for training and 502 for the test set.

To facilitate model selection, we created a validation set consisting of 350 videos which are systematically selected such that the videos represent all the scenarios and the compression types as much as possible (subject to availability of videos). As the VISION data set is not sufficiently large, the validation set could not be fully balanced. It contains a minor data imbalance, which we believe is acceptable for model selection purposes. This resulted in a data set split of 65:35 for (training + validation):test respectively.

Data Set—QUFVD

We also conduct experiments on the newly available Qatar University Forensic Video Database (QUFVD) data set [1]. The data set consists of 20 camera devices such that there are 5 brands, 2 camera models for each brand and 2 identical devices for each camera model. Although the data set does not have the corresponding WhatsApp and YouTube social media versions, this is an interesting data set to test our approach at the device level. The scene content of all the images are natural which helps to simulate real-world scenarios. In comparison to VISION data set, QUFVD contains more recent smartphones. Furthermore, Akbari et al. [1] explicitly divide the data set into train, validation, and test sets. This allows for fair evaluation of the SCI methods on the QUFVD data set.

We conducted our experiments on all the extracted I-frames (these were already provided in the data set) without performing any frame selection. Since the videos in the data set are only a few seconds long, most of the videos have about 11 to 16 I-frames. Overall, there are 192, 42, and 60 videos for train, validation, and test sets, respectively for each camera device and this corresponds to a 80:20 split.

ConvNet Training

We consider multiple ConvNet architectures as described in Sect. 3.2, and evaluate them for camera identification. To perform a fair evaluation between the architectures, we set the same hyperparameters for the learning algorithms, as much as possible. A couple of differences, however, were required and are specified below.

The optimization problem is set up to minimize the categorical cross-entropy loss. We use the stochastic gradient descent (SGD) optimizer with an initial learning rate α (more details below) and momentum of 0.95. A global l2-regularization was included in the SGD optimizer with a decay factor of 0.0005 and a batch size of 64 and 32 for MobileNet and ResNet, respectively. This choice of batch size was based on the limitation of the GPU memory. The ConvNet architectures also includes batch-normalization and dropout layers that aid in model generalization.

Two different sets of hyperparameters were used for the learning of ConvNets. First, we consider the hyperparameters for the experiments not involving constrained convolutions. We set the initial learning rate $\alpha = 0.1$ and employ cosine learning rate decay scheme [37] with three warm-up epochs. Overall we train the system for a total of 20 epochs. The learning rate updates were performed at the end of each batch to ensure a smooth warm-up and decay of the learning rate. The best model was selected based on the epoch which resulted in maximum video-level validation accuracy. In case of a tie between epochs, we select the one with the least validation loss.

A different setting for hyperparameters was used for experiments involving constrained convolutions as proposed by Bayer et al. [6]. We begin with a small learning rate $\alpha = 0.001$ and train for a total of 60 epochs. A step-wise learning rate decay scheme was employed to decay the learning rate by a factor of 2 after every 6 epochs.

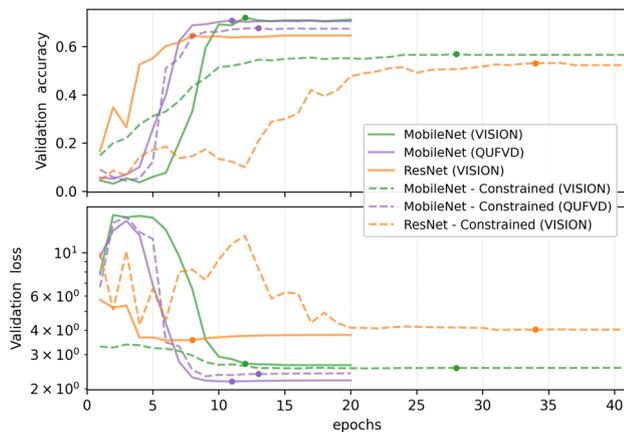


Fig. 4 Epoch-wise accuracy and loss on the validation set for the VISION and the QUFVD data sets. The dot markers indicate the epoch at which the overall best validation accuracy is achieved for the ConvNets

The experiments were conducted on NVIDIA V100 GPUs with 32 GB of video memory. Fig. 4 depicts the convergence plots of the trained models.

ConvNet Evaluation

The ConvNets were evaluated on the test set consisting of 504 videos from 28 camera devices. The evaluation was performed in two phases. We used $N = 50$ I-frames per video to train and test the ConvNet models in the first phase. As described in Sect. 3, 50 frames were extracted from each test video resulting in a total of 25, 200 test frames. In the second phase, we repeated the experiments by not making a selection based on frame type and selecting $N = 50$ frames that are equally spaced in time.

Fifty I-Frames per Video

The training was performed using up to 50 I-frames per video that are equally spaced in time. Note that, when a

video has fewer than 50 I-frames then we considered all the available I-frames for training/evaluation. The trained ConvNets were used to determine the class of each of the 50 I-frames per test video v . These predictions were then aggregated using a majority vote to predict the source camera device for the video v . Having determined predictions for each of the 504 test videos, the overall video-level classification accuracy was determined using:

$$Accuracy = \frac{\# \text{ of correct predictions}}{\text{total } \# \text{ of predictions}} \quad (5)$$

We further investigated the role of the scene suppression techniques as a pre-processing step for the ConvNets. To test this scenario, we trained the ConvNets with a multi-channel constrained convolutional layer as proposed in [52]. In such experiments, the ConvNets were augmented with a constrained convolutional layer. Another popular technique for scene suppression is the extraction of PRNU noise which is built on wavelet based denoising filter proposed in [19, 38]. This has achieved state-of-the-art on camera identification based on images and is used in several research works. Therefore, we also experiment with residual PRNU noise to verify its effectiveness for videos. PRNU noise was extracted from each color channel of the input video frame and the resulting 3-channel PRNU noise inputs were used to train and test the ConvNets. The results of these experiments are reported in Table 2.

On comparing the overall accuracy of all the experiments with 50 I-frames per video, we notice that the unconstrained MobileNet achieves the best accuracy of 72.47 ± 1.1 . This result was obtained after running the same experiment for 5 times and computing the overall average. The ResNet achieves an average accuracy of 67.81 ± 0.5 . On comparing the accuracy of the unconstrained networks to their constrained counterparts [52], we notice that the unconstrained networks perform better by 19.28 and 13.63 percentage points for MobileNet and ResNet, respectively. Moreover, we observe that the traditional technique of scene suppression

Table 2 Classification accuracy of the proposed methods on the VISION data set

| Model | N | Constraint type | Overall | Flat | Indoor | Outdoor | WA | YT | NA |
|-----------|-----|-----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| ResNet50 | 50 | None | 67.81 | 75.96 | 62.60 | 64.24 | 67.76 | 66.94 | 68.82 |
| ResNet50 | 50 | Conv [52] | 54.18 | 61.80 | 50.70 | 40.70 | 51.20 | 50.60 | 60.80 |
| MobileNet | 50 | None | 72.47 | 82.30 | 60.80 | 72.46 | 74.76 | 69.96 | 72.66 |
| MobileNet | 50 | Conv [52] | 53.19 | 60.70 | 45.20 | 52.50 | 52.40 | 48.20 | 59.00 |
| MobileNet | 50 | PRNU [19] | 61.75 | 69.90 | 54.10 | 60.10 | 56.50 | 62.50 | 66.30 |
| MobileNet | all | None | 72.75 | 79.98 | 62.60 | 74.00 | 74.64 | 70.68 | 72.90 |

The third column indicates the type of pre-processing or constraints used on the input video frames. The table presents the overall test accuracy along with the test accuracy for 3 scenarios (flat, indoor, and outdoor) and the 3 compression types (native (NA), WhatsApp (WA), and YouTube (YT)) for all the ConvNets. Furthermore, these results correspond to experiments with N I-frames per video for both training and testing. Best accuracy across experiments for each scenario and compression type are boldfaced

Table 3 Test accuracy of MobileNet when experimenting with N I-frames per video for both the training and test sets on the VISION data set

| N | R1 | R2 | R3 | R4 | R5 | Average |
|-----|------|------|------|------|------|------------------|
| 50 | 72.7 | 73.5 | 73.3 | 71.9 | 70.9 | 72.47 ± 1.07 |
| all | 71.7 | 71.9 | 74.5 | 72.9 | 72.7 | 72.75 ± 1.10 |

The columns R1–R5 indicate the overall accuracy for each of the five runs and the final column shows the respective means and standard deviations

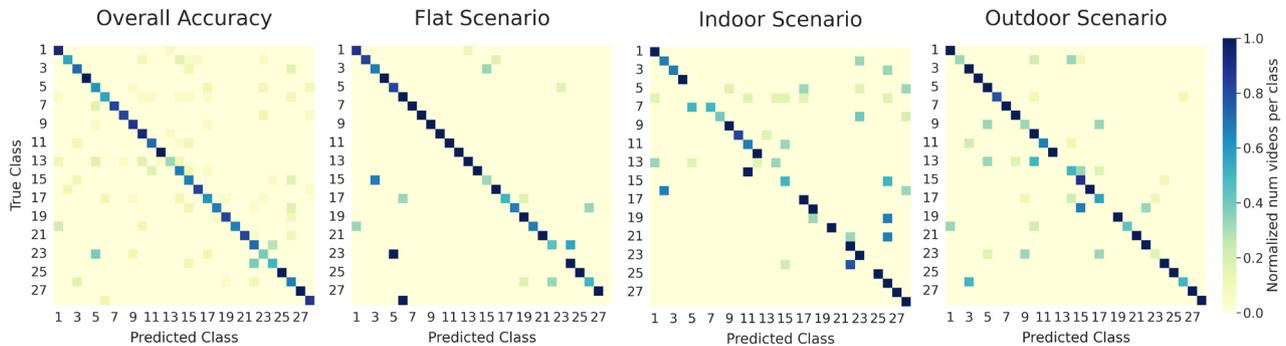


Fig. 5 The confusion matrices obtained by evaluating the MobileNet on the VISION test data set using all I-frames. The overall results along with outcome specific to each of the three scenarios are depicted. The class labels correspond to the sequence of 28 devices listed in Table 1

based on PRNU [19] outperforms the constrained counterparts by a significant margin. Furthermore, on examining the results per scenario we notice that the unconstrained MobileNet and ResNet consistently outperform all other variants. It is interesting to see that the PRNU-based MobileNet comes close in terms of accuracy for native and YouTube scenarios while the performance degrades for WhatsApp videos. This shows that the traditional PRNU based denoising [19] is affected by WhatsApp compression. When considering 50 I-frames per video the unconstrained ResNet performs slightly better than MobileNet for the indoor scenario while the MobileNet performs significantly better in other scenarios.

Instead of limiting to only 50 I-frames per video, we also experimented with all I-frames per video. We considered it beneficial as we have more data, however, we also were aware that it could also cause data imbalance. In the videos considered for our experiments, we observed the number of I-frames per video vary between 8 to 230. The model performs well even in this scenario and the MobileNet achieves an overall accuracy of 72.75%. We further noticed that the results tend to vary between runs, therefore we report the average accuracy across 5 runs for each experiment.

The results, presented in Table 3, indicate that the model is sensitive to a few random components in the network. Firstly, since we are starting from a pre-trained network (model weights learnt on ImageNet), a source of randomness is present in the initial weights of the output layer. In our experiments on the VISION data set, the output layer

contains 28 units in contrast to 1000 units for the ImageNet. The weights of the output layer are initialized using Glorot initialization [18]. Second, the dropout layers can also play a role in contributing towards this randomness. To account for this sensitivity we repeat the experiments for 5 times. We believe the deviations between the runs are pronounced due to the small test set that we have rather than anything to do with the methodology. The best overall accuracy of 74.5 was achieved by the MobileNet when all I-frames were considered in the experiment. In Fig. 5 we illustrate the confusion matrices obtained with the best performing MobileNet.

Comparison with the State-of-the-art

Though the VISION data set consists of both images and videos, most of the works [40] use only images for their analysis. A few works have also been conducted involving videos [28, 39]. Mandelli et al. [39] estimate the reference PRNU noise for a video based on the 50 frames per video extracted from the videos of the flat-still scenario. This approach limits their applicability when flat videos are unavailable for estimating the reference noise pattern \mathbf{K}_v . Iuliani et al. [28] proposed a hybrid solution using 100 images (generated by the same device) to estimate the reference pattern noise \mathbf{K}_{iv} , for stabilized videos and 100 video frames for non-stabilized videos. This approach is again limited by the availability of images from the same device. These works, however, require the knowledge if a video is stabilized beforehand. In practice, the overall classification

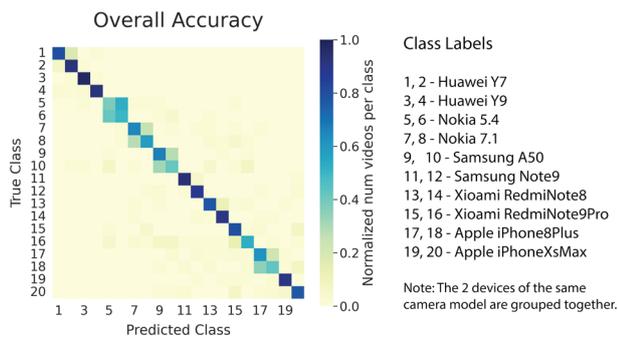


Fig. 6 The confusion matrix obtained by evaluating the MobileNet on the QUFVD test data set using all I-frames. The overall accuracy obtained is 71.75%

accuracy of these methods would therefore be limited by the accuracy of determining the presence of video stabilization.

The recent work of Cortivo et al. [17] included experiments at camera model level with 25 different cameras, trained with only indoor and outdoor samples, and used a data set split of 80:20 for train+val:test. In our experiments, we considered experiments at device level using 28 devices, used all indoor, outdoor and flat scenarios and applied a data set split of 65:35. Cortivo et al. [17] trained three different models, one for each compression type using visual content. In our experiments we combine all compression types and train a single model. This is more practical for forensic investigators when they are unaware of the exact compression type of a given video. These design differences make it unsuitable to have a direct comparison between the two approaches.

To the best of our knowledge, we are the first ones to perform camera identification using videos based on 28 devices (listed in Table 1) from the VISION data set. Furthermore, unlike [28, 39], we ensure that all devices from the same camera model are always included in our data set to test the performance at device level.

We also compare the results obtained using our methodology on the QUFVD data set. We conduct two experiments using MobileNet with and without the constrained convolution layer. The results of these experiments are reported in Table 4. The results indicate that using a pre-trained MobileNet to classify I-frames without any scene-suppression strategy yields better results. Furthermore, we achieve the best result on the QUFVD data set for SCI using all the 20 camera devices. Notable is the fact that for the QUFVD data set the performance of the constrained network is almost on par to that of the unconstrained network. We present the confusion matrix obtained with the unconstrained MobileNet in Fig. 6. It can be seen that, most of the mispredictions are between the devices of the same model.

Table 4 Comparison with Akbari et al. [1] on the QUFVD data set

| Model | Constraint type | Overall accuracy |
|-------------------------|-----------------|------------------|
| [1] MISLNet - grayscale | Conv [6] | 59.60 |
| [1] MISLNet - color | Conv [52] | 51.20 |
| Ours: MobileNet | None | 71.75 |
| Ours: MobileNet | Conv [52] | 70.50 |

All results are reported at the device level

Discussion

Video Compression

It is very common for videos to be shared via social media platforms. This leads to video encoding and compression as per the policy of these platforms. As the VISION data set also includes videos from YouTube and WhatsApp, we investigate the impact of these compressions on the camera identification. The native videos in the data set were shared on YouTube and WhatsApp and both versions were included in the training set during model learning. We independently evaluate the performance of these three compression—native, WhatsApp, and YouTube video versions and report the results in Table 2.

As shown in Table 2, the unconstrained ConvNets perform significantly better when compared to their constrained counterparts. Furthermore, the constrained ConvNets achieve higher accuracy on the native scenarios when compared to WhatsApp and YouTube. Since the native videos contain the unaltered sensor pattern noise, we expect the native video versions to perform better than their compressed counterparts. The results indicate that the extracted features from the sensor pattern noise when encoded into the YouTube and WhatsApp versions still retain most of the camera signatures especially for the unconstrained ConvNets. It is interesting to note that actually for the unconstrained MobileNet, the WhatsApp videos perform slightly better than the native versions. These are promising results and will enable forensic investigators to gain intelligence even from the compressed versions. Note that the sensor pattern noise is partly modified even in the native version as the videos are always generated and stored in compressed formats to save storage.

Number of Frames per Video

In most scenarios, the duration of videos is longer than 10 seconds. Assuming a most common video capture rate of 30 frames per second, we can expect at least 300 frames per video. There could be a few scenarios where the test

Table 5 Classification accuracy for various learning strategies on the VISION data set

| Network | FT | FS | Overall | Flat | Indoor | Outdoor | WA | YT | NA |
|-----------|-----|---------|---------|-------|--------|---------|-------|-------|-------|
| ResNet50 | No | I-frame | 68.13 | 73.40 | 59.60 | 69.90 | 70.20 | 66.70 | 67.50 |
| ResNet50 | Yes | I-frame | 68.53 | 76.30 | 61.00 | 67.20 | 68.50 | 67.90 | 69.30 |
| MobileNet | No | I-frame | 69.52 | 83.20 | 61.00 | 63.40 | 69.00 | 68.50 | 71.10 |
| MobileNet | Yes | I-frame | 73.51 | 81.50 | 61.60 | 75.40 | 77.40 | 69.00 | 74.10 |
| MobileNet | No | Any | 67.13 | 80.30 | 59.60 | 60.70 | 68.50 | 66.10 | 66.90 |
| MobileNet | Yes | Any | 72.51 | 80.90 | 59.60 | 74.90 | 72.60 | 70.20 | 74.70 |

These include fine-tuning (*FT*) and frame selection (*FS*). The results shown in this table correspond to the model which obtained the best overall accuracy across all the runs

Table 6 Test accuracy of MobileNet with different number of frames per video (fpv) on the VISION data set

| # fpv | Overall | Flat | Indoor | Outdoor |
|-------|---------|------|--------|---------|
| 1 | 70.32 | 72.3 | 59.6 | 77.0 |
| 5 | 72.31 | 80.3 | 56.8 | 77.0 |
| 10 | 72.91 | 80.9 | 59.6 | 76.0 |
| 20 | 73.71 | 82.1 | 60.3 | 76.5 |
| 50 | 72.51 | 80.9 | 59.6 | 74.9 |
| 100 | 73.31 | 81.5 | 58.9 | 74.3 |
| 200 | 71.91 | 80.9 | 58.9 | 73.8 |
| 400 | 72.11 | 80.0 | 58.2 | 74.9 |

Table 7 Test accuracy of MobileNet with different number of I-frames per video (I-fpv) on the VISION data set

| # I-fpv | Overall | Flat | Indoor | Outdoor |
|---------|---------|------|--------|---------|
| 1 | 69.12 | 71.1 | 57.5 | 76.5 |
| 5 | 72.31 | 79.8 | 59.6 | 75.4 |
| 30 | 74.10 | 82.1 | 62.3 | 76.0 |
| 50 | 73.51 | 81.5 | 61.6 | 75.4 |
| 100 | 73.71 | 82.1 | 61.6 | 75.4 |
| all | 73.71 | 82.1 | 61.6 | 75.4 |

Since videos may not have the required number of I-frames, we therefore test with up to the specified number of I-fpv

videos can be extremely short. To account for this scenario we train ConvNets with a different strategy for frame selection. Instead of relying on I-frames we now extract 50 frames that are equally spaced in time. This strategy could pick any of the three types of frames (I-, P-, or B-frames). By keeping all learning parameters the same, we train two ConvNets, a MobileNet and a ResNet, and compare their performance with the I-frame counterparts. The results are shown in Table 5. We can see that the I-frames approach is more beneficial than any-frame approach. Furthermore, the close gap between the results indicate that any-frame approach could also be used while encountering videos of very short duration during the training phase.

Additionally, at test time, few videos of extremely short duration may be encountered. We simulate this scheme by testing the performance of our trained model for different number of video frames. In particular, we test the performance on 1, 5, 10, 20, 50, 100, 200, and 400 frames per video that are equally spaced in time, the results of which are presented in Table 6.

As shown in Table 6, MobileNet achieves its best performance when evaluated with 20 frames per video which are equally spaced in time. The difference in overall accuracy between 1 frame per video to 400 frames per video is very small at 1.79 percentage points. Thus, a model trained with 50 frames per video can be expected to perform reasonably well on test videos with very few frames. Interestingly, we

can notice that the performance increases when increasing number of fpv from 1 to 20, however, there is a slight drop in accuracy when considering a large number of frames. The increasing trend in accuracy up to 20 fpv can be attributed to the role of the majority vote. When considering a large number of frames, the majority vote includes frames that are temporally more dense. Thereby, frames that do not result in correct classification could begin to dominate. Thus, using fewer frames that are equally spaced far apart in time are beneficial when performing a simple majority vote.

We conduct a similar experiment using the I-frames, whose results are presented in Table 7. The results indicate that even with very little amount of test I-frames the model achieves a high accuracy. Moreover, as the videos in the VISION data set are of short duration, the number of I-frames are also limited and, therefore, even though we attempt to extract a higher number of I-frames, the accuracy remains the same.

Pre-training ConvNets

In several deep learning tasks based on images, using a pre-trained network improves the classification accuracy of the models. In our experiments, we chose MobileNet and ResNet that are pre-trained on ImageNet. Note that ImageNet is a large-scale object detection data set and the networks pre-trained on ImageNet generalize to learn the high-level scene

details for object recognition. In our setup, we require the ConvNets to extract the low-level noise and therefore it is counter-intuitive to use a pre-trained network that works well to extract high-level scene details. Our experiments show that it is still beneficial to start training from such pre-trained network rather than training from randomly initialized model weights. Table 5 summarizes these experiments. It can be seen that by fine-tuning the overall accuracy for ResNet improves marginally. For MobileNet, however, the overall accuracy improves by 3.99 and 5.38 percentage points for the I-frames and the any-frame approach, respectively.

Camera Model Identification

We test the performance of the trained models for camera model identification. To perform camera model identification, we replace the target device predictions with its corresponding camera model for each of the 502 test videos. The 28 device VISION data set, considered for our experiments, consists of 22 camera models. With this evaluation, we observed an accuracy of 74.70%. This is a marginal improvement over the device level accuracy of 73.51%. A similar evaluation on the QUFVD data set resulted in an improvement in results from 71.75% to 88.5%. In comparison, to the VISION data set this is a significant jump. A network trained directly at model level would perform much better, which can be studied in future work.

Counter-productive Learning Strategies

So far we have elucidated strategies that worked reasonably well for source camera identification for videos. However, it is equally important to discuss strategies that did not work as expected for our problem, which we believe are very beneficial when considering future work.

Majority Voting Scheme

As we are dealing with multiple frames per video it is reasonable to explore the role of various weighted majority voting schemes. The results in Table 2 show that the flat scenarios achieve higher performance when compared to indoor and outdoor scenes. It can be reasoned that since there is no high-level scene content in flat videos, such videos retain higher degree of camera forensic traces. Based on this idea, we quantitatively measured the degree of uniformity by computing homogeneity, entropy and energy based on the gray-level co-occurrence matrix [20]. On appropriately weighing frame predictions with these scores, we did not notice any improvement in accuracy. To give more importance to homogeneous frames we further computed several no reference image quality metrics such as niqe, piqe, and

Table 8 The time duration needed to classify a single video frame for all the ConvNets used in our experiments

| ConvNet | Time length for classifying each frame |
|-------------------------|--|
| MobileNet | 13.92 ms |
| MobileNet - Constrained | 15.62 ms |
| ResNet | 18.95 ms |
| ResNet - Constrained | 18.72 ms |

The measurements were done in milliseconds (ms)

brisque scores [43, 44, 53] but did not notice any patterns that could be exploited for better camera identification.

Frame Selection Schemes

Encouraged by the higher accuracy for flat scenario, we further investigated if such a strategy can be used for frame selection. That is, we determine the homogeneity score of each video frame and only use the top N video frames with the highest homogeneity during prediction. This methodology also did not lead to any improvement.

Scene Suppression Schemes

We already presented our experiments using two different scene-suppression strategies, namely PRNU denoising [19] and the other based on constrained convolutions [6, 52]. When the ConvNets are trained with images of varying scene content but having the same noise signature, the networks are forced to look beyond the high-level scene content and extract the camera noise. In a video, this can be achieved by mixing up of the sequence of video frames and preparing special inputs for the ConvNets. Since green color channel contains twice as much as forensic traces when compared to red and blue color channels [7], we randomly sample 3 different video frames and prepare a new input image with the corresponding green color channels. Such an input image would contain different scene content and same noise signature in the three color channels. With such a strategy we obtained an accuracy of 59.16% using the pre-trained MobileNet.

Prediction Time Efficiency

As demonstrated earlier, making the ConvNets sophisticated resulted in increased accuracy. It is important to ensure that this improvement does not come at a cost of increased computation time. We tested the performance of the trained ConvNets using Intel Xeon CPU E5-2680 and NVIDIA V100 GPU. The performance measurements are listed in Table 8. In contrast, the methods proposed in [39] and [28] take 75

ms and 10 minutes per frame respectively during prediction. Notable is the fact that the performance measurements of the methods can be fairly compared only when evaluated on the same hardware. We report these measurements as stated in the respective works.

In addition to the time it takes to process each video frame, there is a fixed cost that is associated with extraction of frames. This overhead is shared by all video-based SCI methods. In particular, it takes about 26.63 ms to extract a single I-frame from a 720p video, while it takes 42.06 ms for a 1080p video using the ffmpeg library³.

In practice, these numbers translate to a runtime of 202.75 ms to process a 720p video with 5 frames per video. Therefore, 5 videos can be examined in about a second and around 35.5k videos in about 2 hours. In situations where time is a crucial factor for LEAs to collect crucial intelligence, our method can play an important role.

Future Work

As videos share temporal information, one future direction would be to investigate how adjacent frames can be used for scene suppression and enhancing the extraction of camera noise.

The ConvNets used in our experiments, MobileNet-v3-small and ResNet50, were both proposed to process images of size $224 \times 224 \times 3$ pixels. The input resolution that we consider is significantly bigger at $480 \times 800 \times 3$ pixels. We speculate that by scaling these networks further we can extract fine-grained details of the camera noise, leading to improved performance. These strategies can be considered for a future work.

Another future direction is to leverage the structural content of the video file's meta-data. The structural content refers to the individual building blocks and its arrangement that makes up the file's meta-data. Analyzing such non-editable meta-data would augment the information that we currently extract from the visual content of video frames. More insights can therefore be obtained by analyzing a fusion approach that rely on visual content and structural meta-data.

Conclusion

Our approach for camera device identification was designed with the LEAs' practical requirement of high throughput in mind. In fact, our method requires only a relatively small set of video frames to achieve a high accuracy. Such efficiency allows the search space for cameras/videos to be scaled up.

³ <https://www.ffmpeg.org/>

We achieved the best state-of-the-art performance on the QUFVD data set and also demonstrated the effectiveness of our approach on the VISION data set. For the latter data set, a direct comparison, however, was not possible due to several differences in the experimental design.

The unconstrained networks outperform the constrained counterparts for sophisticated networks such as MobileNet and ResNet. It also turned out that using PRNU noise residuals as a means to suppress scene content does not help neither. In fact, this preprocessing step made unconstrained networks less effective. Finally, we analyzed several strategies to determine the aggregated decision obtained from the considered frames, including weighted majority voting, homogeneity based frame selection, and scene suppression strategies for sophisticated ConvNets. The simple majority voting yielded the best results.

The best results for the QUFVD and VISION data sets are achieved with fine tuning a single pre-trained unconstrained MobileNet that takes input from up to 50 I-frames, the outcomes of which are combined by simple majority voting.

Acknowledgements This work was supported by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01. This research has been partly funded with support from the European Commission under the 4NSEEK project with Grant Agreement 821966. This publication reflects the views only of the author, and the European Commission cannot be held responsible for any use which may be made of the information contained therein. We thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high-performance computing cluster.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Akbari Y, Al-maadeed S, Almaadeed N, Al-ali A, Khelifi F, Lawgaly A, et al. A new forensic video database for source smartphone identification: Description and analysis. *IEEE Access* (2022)
2. Alles EJ, Geradts ZJ, Veenman CJ. Source camera identification for heavily JPEG compressed low resolution still images. *J Forensic Sci.* 2009;54(3):628–38.

3. Barni M, Bondi L, Bonettini N, Bestagini P, Costanzo A, Maggini M, Tondi B, Tubaro S. Aligned and non-aligned double JPEG detection using convolutional neural networks. *J Vis Commun Image Represent.* 2017;49:153–63.
4. Barni M, Chen Z, Tondi B.: Adversary-aware, data-driven detection of double JPEG compression: How to make counter-forensics harder. In: 2016 IEEE international workshop on information forensics and security (WIFS), pp. 1–6. IEEE (2016)
5. Bayar B, Stamm M.C. A deep learning approach to universal image manipulation detection using a new convolutional layer. In: Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pp. 5–10 (2016)
6. Bayar B, Stamm MC. Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. *IEEE Trans Inf Forensics Secur.* 2018;13(11):2691–706.
7. Bayer B.E. Color imaging array. United States Patent 3,971,065 (1976)
8. Bayram S, Sencar H, Memon N, Avcibas I. Source camera identification based on CFA interpolation. In: IEEE International Conference on Image Processing 2005, vol. 3, pp. III–69. IEEE (2005)
9. Bennabhaktula S, Alegre E, Karastoyanova D, Azzopardi G.: Device-based image matching with similarity learning by convolutional neural networks that exploit the underlying camera sensor pattern noise. In: In Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods - ICPRAM, pp. 578–584 (2020). <https://doi.org/10.5220/0009155505780584>
10. Bondi L, Baroffio L, Güera D, Bestagini P, Delp EJ, Tubaro S. First steps toward camera model identification with convolutional neural networks. *IEEE Signal Process Lett.* 2016;24(3):259–63.
11. Bondi L, Lameri S, Güera D, Bestagini P, Delp E.J, Tubaro S.: Tampering detection and localization through clustering of camera-based CNN features. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1855–1864. IEEE (2017)
12. Bunk J, Bappy J.H, Mohammed T.M, Nataraj L, Flenner A, Manjunath B, Chandrasekaran S, Roy-Chowdhury A.K, Peterson L.: Detection and localization of image forgeries using resampling features and deep learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1881–1889. IEEE (2017)
13. Cao H, Kot AC. Accurate detection of demosaicing regularity for digital image forensics. *IEEE Trans Inf Forensics Secur.* 2009;4(4):899–910.
14. Chen C, Stamm M.C.: Camera model identification framework using an ensemble of demosaicing features. In: 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6. IEEE (2015)
15. Chuang W.H, Su H, Wu M.: Exploring compression effects for improved source camera identification using strongly compressed video. In: 2011 18th IEEE International Conference on Image Processing, pp. 1953–1956. IEEE (2011)
16. Cozzolino D, Verdoliva L. Noiseprint: a CNN-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security.* 2019;15:144–59.
17. Dal Cortivo D, Mandelli S, Bestagini P, Tubaro S. CNN-based multi-modal camera model identification on video sequences. *J Imaging.* 2021;7(8):135.
18. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
19. Goljan, M., Fridrich, J., Filler, T. Large scale test of sensor fingerprint camera identification. In: Media forensics and security, vol. 7254, pp. 170–181. SPIE (2009)
20. Haralick RM, Shanmugam K, Dinstein IH. Textural features for image classification. *IEEE Trans Syst Man Cybern.* 1973;3(6):610–21.
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
22. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European conference on computer vision, pp. 630–645. Springer (2016)
23. Holst, G.C.: CCD arrays, cameras, and displays. Citeseer (1998)
24. Hosler, B., Mayer, O., Bayar, B., Zhao, X., Chen, C., Shackelford, J.A., Stamm, M.C.: A video camera model identification system using deep learning and fusion. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8271–8275. IEEE (2019)
25. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314–1324 (2019)
26. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017)
27. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141 (2018)
28. Iuliani M, Fontani M, Shullani D, Piva A. Hybrid reference-based video source identification. *Sensors.* 2019;19(3):649.
29. Kang X, Stamm MC, Peng A, Liu KR. Robust median filtering forensics using an autoregressive model. *IEEE Trans Inf Forensics Secur.* 2013;8(9):1456–68.
30. Kharrazi, M., Sencar, H.T., Memon, N.: Blind source camera identification. In: 2004 International Conference on Image Processing, 2004. ICIP'04., vol. 1, pp. 709–712. IEEE (2004)
31. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst.* 2012;25:1097–105.
32. Kurosawa, K., Kuroki, K., Saitoh, N.: Basic study on identification of video camera models by videotaped images. In: Proceedings of 6th Indo Pacific Congress on Legal Medicine and Forensic Sciences, pp. 26–30 (1998)
33. Kurosawa, K., Kuroki, K., Saitoh, N.: CCD fingerprint method-identification of a video camera from videotaped images. In: Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348), vol. 3, pp. 537–540. IEEE (1999)
34. Li CT. Source camera identification using enhanced sensor pattern noise. *IEEE Trans Inf Forensics Secur.* 2010;5(2):280–7.
35. Li J, Li X, Yang B, Sun X. Segmentation-based image copy-move forgery detection scheme. *IEEE Transac Inform Forens Secur.* 2014;10(3):507–18.
36. Lin X, Li CT. Preprocessing reference sensor pattern noise via spectrum equalization. *IEEE Trans Inf Forensics Secur.* 2015;11(1):126–40.
37. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016)
38. Lukas J, Fridrich J, Goljan M. Digital camera identification from sensor pattern noise. *IEEE Trans Inf Forensics Secur.* 2006;1(2):205–14.
39. Mandelli S, Bestagini P, Verdoliva L, Tubaro S. Facing device attribution problem for stabilized video sequences. *IEEE Transac Inform Forens Secur.* 2019;15:14–27.

40. Marra F, Gragnaniello D, Verdoliva L. On the vulnerability of deep learning to adversarial attacks for camera model identification. *Signal Process.* 2018;65:240–8.
41. Marra Francesco, Poggi Giovanni, Sansone Carlo, Verdoliva Luisa. A study of co-occurrence based local features for camera model identification. *Multimedia Tools Appl.* 2017;76(4):4765–81. <https://doi.org/10.1007/s11042-016-3663-0>.
42. Mayer, O., Bayar, B., Stamm, M.C.: Learning unified deep-features for multiple forensic tasks. In: Proceedings of the 6th ACM workshop on information hiding and multimedia security, pp. 79–84 (2018)
43. Mittal A., Moorthy A. K., Bovik A. C. No-reference image quality assessment in the spatial domain. *IEEE Trans Image Process.* 2012;21(12):4695–708. <https://doi.org/10.1109/TIP.2012.2214050>.
44. Mittal A, Soundararajan R, Bovik AC. Making a “completely blind” image quality analyzer. *IEEE Signal Process Lett.* 2012;20(3):209–12.
45. Pibre L, Pasquet J, Ienco D, Chaumont M. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source mismatch. *Electron Imaging.* 2016;2016(8):1–11.
46. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv preprint [arXiv:1710.05941](https://arxiv.org/abs/1710.05941) (2017)
47. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520 (2018)
48. Shullani D, Fontani M, Iuliani M, Al Shaya O, Piva A. VISION: a video and image dataset for source identification. *EURASIP J Inform Secur.* 2017;2017(1):1–16.
49. Swaminathan A, Wu M, Liu KR. Nonintrusive component forensics of visual sensors using output images. *IEEE Trans Inf Forensics Secur.* 2007;2(1):91–106.
50. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114. PMLR (2019)
51. Tang H, Ni R, Zhao Y, Li X. Median filtering detection of small-size image based on CNN. *J Vis Commun Image Represent.* 2018;51:162–8.
52. Timmerman., D., Bennabhaktula., G., Alegre., E., Azzopardi., G.: Video camera identification from sensor pattern noise with a constrained ConvNet. In: Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods - ICPRAM., pp. 417–425. INSTICC, SciTePress (2021). doi:10.5220/0010246804170425
53. Venkatanath, N., Praneeth, D., Bh, M.C., Channappayya, S.S., Medasani, S.S.: Blind image quality evaluation using perception based features. In: 2015 Twenty First National Conference on Communications (NCC), pp. 1–6. IEEE (2015)
54. Wang Xinyi, Wang He, Niu Shaozhang. An image forensic method for AI inpainting using faster R-CNN. In: Sun Xingming, Pan Zhaoqing, Bertino Elisa, editors. Artificial intelligence and security: 5th International Conference, ICAIS 2019, New York, NY, USA, July 26–28, 2019, Proceedings, Part III. Cham: Springer International Publishing; 2019. p. 476–87. https://doi.org/10.1007/978-3-030-24271-8_43.
55. Zhu X, Qian Y, Zhao X, Sun B, Sun Y. A deep learning approach to patch-based image inpainting forensics. *Signal Process.* 2018;67:90–9.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.